

Практическое занятие № 15

Наименование практического занятия: составление программ для работы с базами данных в IDE PyCharm Community.

Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

Вариант № 9

Задание:

Средствами Python реализовать программу для работы с однотабличной БД. Программа должна обеспечивать функционал по вводу данных в БД (10 позиций), их поиску, удалению и редактированию. При организации поиска, удаления и редактирования использовать WHERE, предусмотреть по три SQL-запроса для каждой операции. Структура БД указана в каждом варианте.

Приложение УЧЕБНЫЙ ПЛАН для автоматизированного контроля учебной нагрузки по кафедре. Таблица Дисциплины должна иметь следующую структуру записи: Код дисциплины, Наименование дисциплины, Специальность, Лекции (колич.часов), Практические (колич.часов), Лабораторные (колич.часов), Форма отчетности.

Код:

```
import sqlite3 as sq
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QTableWidget, QTableWidgetItem, QTableWidgetItem, QTableWidget
from PyQt5.QtCore import Qt

def create_table():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        cur.execute('''
            CREATE TABLE IF NOT EXISTS "Disciplines" (
                "Discipline_code" INTEGER UNIQUE,
                "Discipline_name" TEXT,
                "Speciality" TEXT,
                "Lectures_number_of_hours" INTEGER,
                "Practical_number_of_hours" INTEGER,
                "Laboratory_number_of_hours" INTEGER,
                "Reporting_form" TEXT,
                PRIMARY KEY("Discipline_code" AUTOINCREMENT)
            )
        ''')
```

```

def insert_data():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        cur.execute('''
            INSERT OR IGNORE INTO "Disciplines"
                ("Discipline_name", "Speciality", "Lectures_number_of_hours",
                 "Practical_number_of_hours", "Laboratory_number_of_hours",
                 "Reporting_form")
            VALUES
                ("Математика", "ИС", 48, 32, 0, "очная"),
                ("физика", "ИС", 32, 32, 0, "заочная"),
                ("Программирование", "ПОКС", 32, 32, 32, "очная"),
                ("История", "ИКС", 32, 0, 0, "очная"),
                ("Литература", "ИС", 32, 0, 0, "очная"),
                ("Иностранный язык", "ИС", 32, 0, 0, "очная")
        ''')

def read_data():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        cur.execute('SELECT * FROM "Disciplines"')
        rows = cur.fetchall()
        return rows

def update_record():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        discipline_code = input('Введите код дисциплины, которую хотите изменить: ')
        new_discipline_name = input('Введите новое название дисциплины: ')
        new_speciality = input('Введите новую специальность: ')
        new_lectures = input('Введите новое количество лекций: ')
        new_practical = input('Введите новое количество практических занятий: ')
        new_laboratory = input('Введите новое количество лабораторных работ: ')
        new_reporting_form = input('Введите новую форму отчетности: ')

        cur.execute('''
            UPDATE "Disciplines" SET
                "Discipline_name" = ?,
                "Speciality" = ?,
                "Lectures_number_of_hours" = ?,
                "Practical_number_of_hours" = ?,
                "Laboratory_number_of_hours" = ?,
                "Reporting_form" = ?
            WHERE "Discipline_code" = ?
        ''', (new_discipline_name, new_speciality, new_lectures, new_practical,
              new_laboratory, new_reporting_form, discipline_code))
        con.commit()

def delete_record():
    with sq.connect('PZ15/pz15_DB.db') as con:

```

```

        cur = con.cursor()
        discipline_code = input('Введите код дисциплины, которую хотите удалить: ')
        cur.execute('DELETE FROM "Disciplines" WHERE "Discipline_code" = ?',
        (discipline_code,))
        con.commit()

def insert_new_record():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        new_discipline_name = input('Введите название новой дисциплины: ')
        new_speciality = input('Введите специальность новой дисциплины: ')
        new_lectures = input('Введите количество часов лекций новой дисциплины: ')
        new_practical = input('Введите количество часов практических занятий новой
дисциплины: ')
        new_laboratory = input('Введите количество часов по лабораторным работам
новой дисциплины: ')
        new_reporting_form = input('Введите форму отчетности новой дисциплины: ')
        cur.execute('''
            INSERT INTO "Disciplines"
                ("Discipline_name", "Speciality", "Lectures_number_of_hours",
                "Practical_number_of_hours", "Laboratory_number_of_hours",
                "Reporting_form")
            VALUES
                (?, ?, ?, ?, ?, ?)
        ''', (new_discipline_name, new_speciality, new_lectures, new_practical,
        new_laboratory, new_reporting_form))
        con.commit()

def table_is_empty():
    with sq.connect('PZ15/pz15_DB.db') as con:
        cur = con.cursor()
        cur.execute('SELECT COUNT(*) FROM "Disciplines"')
        return cur.fetchone()[0] == 0

create_table()

if table_is_empty():
    insert_data()

class MyTable(QWidget):
    def __init__(self, data):
        super().__init__()

        self.table = QTableWidgetItem()
        self.table.setRowCount(len(data))
        self.table.setColumnCount(len(data[0]))
        self.table.setHorizontalHeaderLabels(
            ["Код дисциплины", "Наименование дисциплины", "Специальность", "Лекции
(колич.часов)",
            "Практические (колич.часов)", "Лабораторные (колич.часов)", "Форма
отчетности"])

```

```

        for i, row in enumerate(data):
            for j, item in enumerate(row):
                self.table.setItem(i, j, QTableWidgetItem(str(item)))

        layout = QVBoxLayout()
        layout.addWidget(self.table)
        self.setLayout(layout)
        self.setFixedSize(925, 600)

def show_table():
    data = read_data()
    app = QApplication(sys.argv)
    table = MyTable(data)
    table.show()
    sys.exit(app.exec_())

while True:
    print("-" * 42)
    print('| Выберите действие: ' , " " * 19 , "|")
    print("-" * 42)
    print('| 1. Просмотреть таблицу' , " " * 15 , "|")
    print("-" * 42)
    print('| 2. Изменить строку' , " " * 19 , "|")
    print("-" * 42)
    print('| 3. Удалить строку' , " " * 20 , "|")
    print("-" * 42)
    print('| 4. Добавить новую строку' , " " * 13 , "|")
    print("-" * 42)
    print('| 5. Выйти' , " " * 29 , "|")
    print("-" * 42)

    choice = input('*Введите номер действия -> ')

    if choice == '1':
        show_table()
    elif choice == '2':
        update_record()
    elif choice == '3':
        delete_record()
    elif choice == '4':
        insert_new_record()
    elif choice == '5':
        break
    else:
        print('Неверный номер действия. Повторите ввод.')

```

Протокол выполнения программы:

1. Пользователь выбирает один из пяти режимов работы программы

1. Просмотреть таблицу
2. Изменить строку
3. Удалить строку
4. Добавить новую строку
5. Выйти

2. Программа выполняет заданное пользователем действие с помощью функций:

1. Show_table()
2. Update_record()
3. Delete_record()
4. Insert_new_record()
5. Выход из меню осуществляется с помощью конструкции “break”

3. Вывод необходимых данных с помощью input

discipline_code = input('Enter the code of the discipline you want to change: ')

new_discipline_name = input('Enter a new discipline name: ')

new_specialty = input('Enter a new specialty: ')

new_lectures = input('Enter the new number of lectures: ')

new_practical = input('Enter the new number of practical lessons: ')

new_laboratory = input('Enter the new number of laboratory works: ')

new_reporting_form = input('Enter a new reporting form: ')

... И Т. П.

4. Вывод результата работы программы.

Вывод программы:

```
-----
| Выберите действие: |
|-----|
| 1. Просмотреть таблицу |
|-----|
| 2. Изменить строку |
|-----|
| 3. Удалить строку |
|-----|
| 4. Добавить новую строку |
|-----|
| 5. Выйти |
|-----|
*Введите номер действия -> █
```

python

	Код дисциплины	название дисциплины	Специальность	лекции (колич.час)	практические (колич.ч)	лабораторные (колич.)	Форма отчетности
1	1	Математика	ИС	48	32	0	очная
2	2	физика	ИС	32	32	0	заочная
3	3	Программирова...	ПОКС	32	32	32	очная
4	4	История	ИКС	32	0	0	очная
5	5	Литература	ИС	32	0	0	очная
6	6	Иностранный ...	ИС	32	0	0	очная

Вывод:

Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

P.S

В работе использовались сторонние библиотеки которые не представлены в задании (по согласованию с преподавателем)